8

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/547,288	04/11/2000	Nir N. Shavit	1004-4663	4871

22120 7590 04/07/2003

ZAGORIN O'BRIEN & GRAHAM LLP
401 W 15TH STREET
SUITE 870
AUSTIN, TX 78701

EXAMINER	
LI, AIMEE J	
ART UNIT	**PAPER NUMBER**
2183	7

DATE MAILED: 04/07/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 07-01)

Office Action Summary	Application No.	Applicant(s)	
	09/547,288	SHAVIT ET AL.	
	Examiner	Art Unit	
	Aimee J Li	2183	

-- The MAILING DATE of this communication appears on the c ver sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11 April 2000 and 02 April 2002*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-43* is/are pending in the application.

4a) Of the above claim(s) ____ is/are withdrawn from consideration.

5)☐ Claim(s) ____ is/are allowed.

6)☒ Claim(s) *1-43* is/are rejected.

7)☒ Claim(s) *37* is/are objected to.

8)☐ Claim(s) ____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on ____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on ____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. ____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☒ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) 3, ~~7~~ 5, 6.
4) ☐ Interview Summary (PTO-413) Paper No(s). ____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other:

DETAILED ACTION

1. Claims 1-43 have been considered.

Claim Objections

2. Claim 37 is objected to because of the following informalities: Please correct the phrase "opposing end variants of the push operation" on line 5 to read --opposing end variants of the pop operation--. Appropriate correction is required.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-9, 15-24, and 32-41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mark Allen Weiss's Data Structures and Algorithm Analysis in C++ Second Edition © 1999 (herein referred to as Weiss) in view of Arnold, EPO 0366585 A2 (herein referred to as Arnold).

5. Referring to claim 1, Weiss has taught a method of managing access to an array susceptible to concurrent operations on a sequence encoded therein, the method comprising:

- a. Executing as part of a pop operation, an operation to atomically update a then-current, end identifying index for the array and a element of the array adjacent to that identified by the end identifying index (Weiss pages 110-114)
- b. Returning from the operation, on failure thereof, an indication by which an empty state of the array is detectable (Weiss pages 72 and 110-114).

6. Weiss has not taught a double compare and swap (DCAS) as the specific operation. Arnold has taught using a DCAS as part of a push or pop operation (Arnold pages 2-3). It would have been obvious to a person of ordinary skill in the art to incorporate the DCAS operation of Arnold, because the DCAS would eliminate the need for a locking mechanism, which degrades performance, and allows multiple processes to access the shared data at once. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the DCAS of Arnold in the device of Weiss to improve processing performance.

7. Referring to claim 2, Weiss has taught wherein the indication by which the empty state of the array is detectable is indicative of presence of a distinguishing value in the adjacent element (Weiss pages 110-114).

8. Referring to claim 3, Weiss has taught wherein the array encodes a double-ended queue as a circular buffer of bounded size, the end identifying index and an opposing end identilying index delimiting the sequence (Weiss pages 110-114)

9. Referring to claim 4, Weiss has taught:

a. Wherein the pop operation is a left pop operation (Weiss pages 110-114)

b. Wherein the end identifying index is a left-end index (Weiss pages 110-114)

c. Wherein the adjacent element is to the right of the identified element (Weiss pages 110-114).

10. Referring to claim 5, Weiss has taught:

a. Wherein the pop operation is a right pop operation (Weiss pages 110-114)

b. Wherein the end identifying index is a right-end index (Weiss pages 110-114)

c. Wherein the adjacent element is to the left of the identified element (Weiss pages 110-114).

11. Referring to claim 6, Weiss has taught a method of managing access to an array susceptible to concurrent operations on a sequence encoded therein, the method comprising:

a. Executing as part of a push operation, an operation to atomically update a then-current, end identiiying index for the array and an element of the array identified by the end identifying index (Weiss pages 110-114)

b. Returning from the operation, on failure thereof, an indication by which a full state of the array is detectable (Weiss pages 72 and 110-114).

12. Weiss has not taught a double compare and swap (DCAS) as the specific operation. Arnold has taught using a DCAS as part of a push or pop operation (Arnold pages 2-3). It would have been obvious to a person of ordinary skill in the art to incorporate the DCAS operation of Arnold, <u>because</u> the DCAS would eliminate the need for a locking mechanism, which degrades performance, and allows multiple processes to access the shared data at once. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the DCAS of Arnold in the device of Weiss to improve processing performance.

13. Referring to claim 7, Weiss has taught wherein the indication by which the full state of the array is detectable is indicative of absence of a distinguishing value in the identified element (Weiss pages 110-114).

14. Referring to claim 8, Weiss has taught:

a. Wherein the push operation is a left push operation (Weiss pages 110-114)

b. Wherein the end identifying index is a left-end index (Weiss pages 110-114).

15. Referring to claim 9, Weiss has taught:

a. Wherein the push operation is a right push operation (Weiss pages 110-114)

b. Wherein the end identifying index is a right-end index (Weiss pages 110-114).

16. Referring to claim 15, Weiss has taught a method of managing concurrent access to a double-ended queue (deque), the method comprising:

a. Employing, in an implementation of a pop operation, execution of an operation to interrogate instantaneous values of a first end index and a deque element adjacent to that identified thereby for a signature indicative of an empty state of the array, the signature including presence in that adjacent element of a distinguishing value (Weiss pages 110-114)

b. Wherein successful execution of an opposing end pop operation includes ' execution of an operation to atomically update a second end index and a deque element adjacent to that identified thereby, the update of that adjacent element storing the distinguishing value therein (Weiss pages 110-114)

17. Weiss has not taught a double compare and swap (DCAS) as the specific operation. Arnold has taught using a DCAS as part of a push or pop operation (Arnold pages 2-3). It would have been obvious to a person of ordinary skill in the art to incorporate the DCAS operation of Arnold, <u>because</u> the DCAS would eliminate the need for a locking mechanism, which degrades performance, and allows multiple processes to access the shared data at once. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was

made to incorporate the DCAS of Arnold in the device of Weiss to improve processing performance.

18. Referring to claim 16, Weiss has taught the method further comprising wherein successful execution of a competing, same end pop operation includes execution of an operation to atomically update the first end index and a deque element adjacent to that identified thereby, the update of that adjacent element storing the distinguishing value therein. Weiss has not taught a double compare and swap (DCAS) as the specific operation. Arnold has taught using a DCAS as part of a push or pop operation (Arnold pages 2-3). It would have been obvious to a person of ordinary skill in the art to incorporate the DCAS operation of Arnold, <u>because</u> the DCAS would eliminate the need for a locking mechanism, which degrades performance, and allows multiple processes to access the shared data at once. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the DCAS of Arnold in the device of Weiss to improve processing performance.

19. Referring to claim 17, Weiss has taught:

a. Wherein the first end index is a left index and, if the state of the deque is nonempty, the deque element adjacent to that identified thereby is a left most element of the deque (Weiss pages 79-80 and 110-114);

b. Wherein the second end index is a right index and, if the state of the deque is non-empty, the deque element adjacent to that identified thereby is the right most element of the deque (Weiss pages 79-80 and 110-114).

20. Referring to claim 18, Weiss has taught:

a. Wherein the pop operation is a left pop operation and the opposing end pop operation is a right pop operation (Weiss pages 110-114); and

b. Wherein the first end index is a left end index and the element adjacent to that identified thereby is adjacent to the right (Weiss pages 110-114).

21. Referring to claim 19, Weiss has taught wherein the distinguishing value is encoded as a null value (Weiss pages 71-72).

22. Referring to claim 20, Weiss has taught the method further comprising:

a. Employing, in an implementation of a push operation, execution of an operation to interrogate instantaneous values of a third end index and a deque element identified thereby for a signature indicative of an flill state of the deque, the signature including absence in that identified deque element of a distinguishing value (Weiss pages 110-114),

b. Wherein successful execution of an opposing end push operation includes execution of an operation to atomically update a fourth end index and a deque element identified thereby, the update of the identified deque element storing a value other than the distinguishing value therein (Weiss pages 110-114).

23. Weiss has not taught a double compare and swap (DCAS) as the specific operation. Arnold has taught using a DCAS as part of a push or pop operation (Arnold pages 2-3). It would have been obvious to a person of ordinary skill in the art to incorporate the DCAS operation of Arnold, <u>because</u> the DCAS would eliminate the need for a locking mechanism, which degrades performance, and allows multiple processes to access the shared data at once. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was

made to incorporate the DCAS of Arnold in the device of Weiss to improve processing performance.

24. Referring to claim 21, Weiss has taught:

a. Wherein the first end index and the third end index identify a same end of the deque (Weiss pages 110-114); and

b. Wherein the second end index and the fourth end index identify a same end of the deque (Weiss pages 110-114).

25. Referring to claim 22, Weiss has taught:

a. Wherein the first end index and the fourth end index identify a same end of the deque (Weiss pages 110-114); and

b. Wherein the second end index and the third end index identify a same end of the deque (Weiss pages 110-114).

26. Referring to claim 23, Weiss has taught a method of managing concurrent access to a double-ended queue (deque), the method comprising:

a. Employing, in an implementation of a push operation, execution of an operation to interrogate instantaneous values of a first end index and a deque element identified thereby for a signature indicative of a full state of the deque, the signature including absence in that identified deque element of a distinguishing value (Weiss pages 110-114),

b. Wherein successful execution of an opposing end push operation includes execution of an operation to atomically update an opposing end index and a deque

element identified thereby, the update of the' identified deque element storing a value other than the distinguishing value therein (Weiss pages 110-114).

27. Weiss has not taught a double compare and swap (DCAS) as the specific operation. Arnold has taught using a DCAS as part of a push or pop operation (Arnold pages 2-3). It would have been obvious to a person of ordinary skill in the art to incorporate the DCAS operation of Arnold, <u>because</u> the DCAS would eliminate the need for a locking mechanism, which degrades performance, and allows multiple processes to access the shared data at once. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the DCAS of Arnold in the device of Weiss to improve processing performance.

28. Referring to claim 24, Weiss has taught the method further comprising wherein successful execution of a competing, same end push operation includes execution of a DCAS to atomically update the first end index and a deque element identified thereby, the update of that adjacent element storing a value other than the distinguishing value therein (Weiss pages 110-114).

29. Referring to claims 32-35, Weiss has taught a double-ended queue (deque) implementation comprising:

a. A contiguous array S of bounded size encoded in an addressable store (Applicant's claim 32) (Weiss pages 110-114);

b. A left index L and a right index R into the contiguous array, the contiguous array S, the left index L and the right index R together defining a circular buffer with state including a sequence of zero or more values encoded in the contiguous array

between elements S[L] and S[R] thereof (Applicant's claim 32) (Weiss pages 110-114)

c. A computer readable encoding of at least a first access operation, execution of the first access operation operating at a particular end of the sequence and employing an operation to atomically update a corresponding one, but not both, of the left and right indices L and R and an element of the contiguous array adjacent to the contiguous array element identified thereby (Applicant's claim 32) (Weiss pages 110-114).

d. Wherein the first access operation includes a push (Applicant's claim 33) (Weiss pages 110-114)

e. Wherein, on failure, the operation returns an indication by which a full state of the contiguous array is detected (Applicant's claim 33) (Weiss pages 110-114).

f. Wherein the first access operation includes a pop (Applicant's claim 34) (Weiss pages 110-114)

g. Wherein, on failure, the operation returns an indication by which an empty state of the contiguous array is detected (Applicant's claim 34) (Weiss pages 110-114).

h. Computer readable encodings of at least three additional access operations (Applicant's claim 35) (Weiss pages 110-114),

i. Wherein the first and the three additional access operations together include push and pop operations at left and rights end of the sequence, respectively (Applicant's claim 35) (Weiss pages 110-114).

30. Weiss has not taught a double compare and swap (DCAS) as the specific operation. Arnold has taught using a DCAS as part of a push or pop operation (Arnold pages 2-3). It would have been obvious to a person of ordinary skill in the art to incorporate the DCAS operation of Arnold, because the DCAS would eliminate the need for a locking mechanism, which degrades performance, and allows multiple processes to access the shared data at once. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the DCAS of Arnold in the device of Weiss to improve processing performance.

31. Referring to claim 36, Weiss has taught a concurrent shared object implementation comprising:

a. A contiguous array encoded in an addressable store (Weiss pages 110-114);

b. Opposing indices into the contiguous array usable to delimit therebetween a portion of the contiguous array for storage of a sequence of zero or more data values (Weiss pages 110-114)

c. A computer readable encoding of push and pop operations defined to operate on elements of the contiguous array and on respective of the opposing indices (Weiss pages 110-114),

d. Wherein the push operation employs a first instance of an operation to atomically update one of the opposing indices and a corresponding element of the contiguous array while returning on failure, an indication by which a full state of the contiguous array is detected (Weiss pages 110-114), and

e. Wherein the pop operation employs a second instance of an operation to atomically update one of the opposing indices and a corresponding element of the contiguous array while returning on failure, an indication by which an empty state of the contiguous array is detected (Weiss pages 110-114).

32. Weiss has not taught a double compare and swap (DCAS) as the specific operation. Arnold has taught using a DCAS as part of a push or pop operation (Arnold pages 2-3). It would have been obvious to a person of ordinary skill in the art to incorporate the DCAS operation of Arnold, because the DCAS would eliminate the need for a locking mechanism, which degrades performance, and allows multiple processes to access the shared data at once. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the DCAS of Arnold in the device of Weiss to improve processing performance.

33. Referring to claim 37, Weiss has taught:

a. Wherein concurrent shared object includes a deque (Weiss pages 110-114); and

b. Wherein the computer readable encoding of push and pop operations includes: opposing end variants of the push operation; and opposing end variants of the push operation (Weiss pages 110-114).

34. Referring to claim 38, Weiss has taught:

a. Wherein concurrent shared object includes a queue or FIFO (Weiss pages 110-114); and

b. Wherein the computer readable encoding of push and pop operations operate on opposing ends of the queue or FIFO (Weiss pages 110-114).

35. Referring to claim 39, Weiss has taught:

a. Wherein concurrent shared object includes a stack or LIFO (Weiss page 93); and

b. Wherein the computer readable encoding of push and pop operations operate on a same end of the stack or LIFO (Weiss pages 93-100).

36. Referring to claim 40, Weiss has taught a computer program product encoded in at least one computer readable medium, the computer program product comprising:

a. At least one functional sequence implementing an access operation on a concurrent shared object, the concurrent shared object instantiable circular buffer of bounded size implementing a contiguous array delimited by a pair of end identifying indices (Weiss pages 110-114);

b. Instances of the at least one functional sequence concurrently executable by plural processors of a multiprocessor and each including an operation to atomically update a corresponding one of the end identifying indices and an element of the array corresponding to a then-current value thereof (Weiss pages 110-114); and

c. The operation of the at least one functional sequence responsive to a corresponding boundary condition state of the concurrent shared object (Weiss pages 110-114).

37. Weiss has not taught a double compare and swap (DCAS) as the specific operation. Arnold has taught using a DCAS as part of a push or pop operation (Arnold pages 2-3). It would have been obvious to a person of ordinary skill in the art to incorporate the DCAS operation of Arnold, <u>because</u> the DCAS would eliminate the need for a locking mechanism, which degrades performance, and allows multiple processes to access the shared data at once. Therefore, it

would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the DCAS of Arnold in the device of Weiss to improve processing performance.

38. Referring to claim 41, Weiss has taught:

a. Wherein the at least one functional sequence includes opposing end variants of push and pop operations on the concurrent shared object (Weiss pages 110-114);

b. Wherein the boundary condition state corresponding to push operations is a full state of the array (Weiss pages 110-114); and

c. Wherein the boundary condition state corresponding to pop operations is an empty state of the array (Weiss pages 110-114).

39. Claim 42 is rejected under 35 U.S.C. 103(a) as being unpatentable over Mark Allen Weiss's <u>Data Structures and Algorithm Analysis in C++ Second Edition</u> © 1999 (herein referred to as Weiss) in view of Arnold, EPO 0366585 A2 (herein referred to as Arnold) as applied to claims 40-41 above, and further in view of David A. Patterson and John L. Hennessy's <u>Computer Architecture A Quantitative Approach Second Edition</u> ©1996 (herein referred to as Hennessy). Weiss has not taught wherein the at least one computer readable medium is selected from the set of a disk, tape or other magnetic, optical, or electronic storage medium and a network, wireline, wireless or other communications medium. Hennessy has taught wherein the at least one computer readable medium is selected from the set of a disk, tape or other magnetic, optical, or electronic storage medium and a network, wireline, wireless or other communications medium (Hennessy pages 485-495 and 562-572). It would have been obvious to a person of ordinary skill in the art to incorporate the computer readable mediums of Hennessy, <u>because</u> the data and

processes must be stored somewhere accessible by the computer for use. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the computer readable mediums of Hennessy in the device of Weiss.

40. Claims 10-14, 25-31, and 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Arnold, EPO 0366585 A2 (herein referred to as Arnold) in view of Douglas Comer's Operating System Design: The Xino Approach ©1984 (herein referred to as Comer).

41. Referring to claim 10, Arnold has taught a method of providing concurrent access to a double-ended data structure of bounded size implemented using a circular buffer technique, the method comprising:

a. As part of an access to a first-end of the double-ended data structure, performing in alternate legs of a conditional branch:

   i. A first multi-way compare and swap on then-current contents of a first-end index store and a corresponding element of the double-ended data structure to disambiguate a retry state and a state of the double-ended data structure (Arnold pages 2-3);

   ii. A second multi-way compare and swap on then-current contents of the first-end index store and a corresponding element of the double-ended data structure, the second multi-way compare and swap performing the access and, on failure thereof, returning an indication disambiguating a retry state and the state of the double-ended data structure (Arnold pages 2-3),

b. Wherein the conditional blanch discriminates between presence and absence of a distinguishing value in an element of the double-ended data structure

corresponding to the then-current contents of the first-end index store (Arnold pages 2-3).

42. Arnold has not taught detecting a boundary condition state of the double-ended data structure. Comer has taught detecting a boundary condition state of the double-ended data structure (Comer pages 41-42). It would have been obvious to a person of ordinary skill in the art to incorporate the detection of Comer, because in a double-ended data structure boundary conditions must be monitored so an error does not occur. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the detection of Comer in Arnold to prevent errors.

43. Referring to claim 11, Arnold has taught:

a. Wherein the access includes a pop from the first-end of the double-ended data structure (Arnold page 2, lines 26-27);

b. Wherein the retry state results from a concurrently performed push or pop access at the first-end of the double-ended data structure (Arnold pages 2-3).

44. Arnold has not taught wherein the boundary condition state is an empty state of the double-ended data structure. Comer has taught wherein the boundary condition state is an empty state of the double-ended data structure (Comer pages 41-43). It would have been obvious to a person of ordinary skill in the art to incorporate the boundary condition state of Comer, because when this condition is not treated separately, errors occur. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the boundary condition of Comer in the device of Arnold.

45. Referring to claim 12, Arnold has taught:

a. Wherein the access includes a push onto the first-end of the double-ended data structure (Arnold page 2, lines 26-27);

b. Wherein the retry state results from a concurrently performed push or pop access at the first-end of the double-ended data structure (Arnold pages 2-3).

46. Arnold has not taught wherein the boundary condition state is a full state of the double-ended data structure. Comer has taught wherein the boundary condition state is a full state of the double-ended data structure (Comer pages 41-43). It would have been obvious to a person of ordinary skill in the art to incorporate the boundary condition state of Comer, <u>because</u> when this condition is not treated separately, errors occur. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the boundary condition of Comer in the device of Arnold.

47. Referring to claim 13, Arnold has taught wherein the double-ended data structure includes a double-ended queue (deque) (Arnold page 2, lines 5-7).

48. Referring to claim 14, wherein the multi-way compare and swap is a double compare and swap (DCAS) (Arnold pages 2-3).

49. Referring to claim 25, Arnold has taught a method of managing concurrent access to an array susceptible to competing accesses at same and opposing ends thereof, the method comprising:

a. Executing as part of a first access operation, a double compare and swap (DCAS) to atomically update a first end identifying index and an element of the array corresponding to a then-current value thereof;

b. Executing as part of a competing second access operation, a DCAS to atomically update a second end identifying index and an element of the array corresponding to a then-current value thereof,

c. Wherein, if successful completion of one of the first and the second competing access operations results in a certain state of the array, the DCAS of the other of the first and the second access operations fails and returns an indication thereof.

50. Arnold has not taught a boundary condition state as the certain state of the array. Comer has taught a boundary condition state of the array (Comer pages 41-42). It would have been obvious to a person of ordinary skill in the art to incorporate the detection of Comer, <u>because</u> in a double-ended data structure boundary conditions must be monitored so an error does not occur. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the detection of Comer in Arnold to prevent errors.

51. Referring to claim 26, Arnold has taught:

a. Wherein the first access operation and the competing second access operation are competing pop operations (Arnold page 2, lines 26-27).

b. Wherein the adjacent element referenced by the failing one of the competing pop operations encodes a distinguishing value signifying the empty state (Arnold pages 2-3).

52. Arnold has not taught:

a. Wherein the array elements corresponding to the first and second indices are each adjacent to that identified by the respective index

b. Wherein the boundary condition state is an empty state

53. Comer has taught:

a. Wherein the array elements corresponding to the first and second indices are each adjacent to that identified by the respective index (Comer pages 41-43)

b. Wherein the boundary condition state is an empty state (Comer pages 41-43)

54. It would have been obvious to a person of ordinary skill in the art to incorporate the array of Comer, because it is necessary to track the beginning and end of an array as well as the bound condition states of the array to ensure correct execution of push and pop operations. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the array of Comer in the device of Arnold.

55. Referring to claim 27, Arnold has taught wherein the competing pop operations are competing opposing end pop operations (Arnold page 2, lines 26-27). Arnold has not taught wherein the first index and the second index identify opposing ends of the array. Comer has taught wherein the first index and the second index identify opposing ends of the array (Comer pages 41-43). It would have been obvious to a person of ordinary skill in the art to incorporate the first index and second index of Comer, because the beginning and end of the array must be tracked in order not to lose the array and to know where to insert and remove elements. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the first index and second index of Comer in the device of Arnold.

56. Referring to claim 28, Arnold has taught wherein the competing pop operations are competing same end pop operations (Arnold page 2, lines 26-27). Arnold has not taught wherein the first index and the second index identify a same end of the array. Comer has taught wherein

the first index and the second index identify a same end of the array (Comer pages 41-43). It would have been obvious to a person of ordinary skill in the art to incorporate the first index and second index of Comer, because the beginning and end of the array must be tracked in order not to lose the array and to know where to insert and remove elements. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the first index and second index of Comer in the device of Arnold.

57. Referring to claim 29, Arnold has taught:

  a. Wherein the first access operation and the competing second access operation are competing push operations (Arnold page 2, lines 26-27);

  b. Wherein the array element referenced by the failing one of the competing push operations encodes a value other than a distinguishing value (Arnold pages 2-3).

58. Arnold has not taught:

  a. Wherein the array elements corresponding to the first and second indices are each identified by the respective index;

  b. Wherein the boundary condition state is an full state; and

59. Comer has taught:

  a. Wherein the array elements corresponding to the first and second indices are each identified by the respective index (Comer pages 41-43);

  b. Wherein the boundary condition state is an full state (Comer pages 41-43);

60. It would have been obvious to a person of ordinary skill in the art to incorporate the array of Comer, because it is necessary to track the beginning and end of an array as well as the bound condition states of the array to ensure correct execution of push and pop operations. Therefore, it

would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the array of Comer in the device of Arnold.

61. Referring to claim 30, Arnold has taught wherein the competing push operations are competing opposing end push operations (Arnold page 2, lines 26-27). Arnold has not taught wherein the first index and the second index identify opposing ends of the array. Comer has taught wherein the first index and the second index identify opposing ends of the array (Comer pages 41-43). It would have been obvious to a person of ordinary skill in the art to incorporate the first index and second index of Comer, <u>because</u> the beginning and end of the array must be tracked in order not to lose the array and to know where to insert and remove elements. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the first index and second index of Comer in the device of Arnold.

62. Referring to claim 31, Arnold has taught wherein the competing push operations are competing same end push operations (Arnold page 2, lines 26-27). Arnold has not taught wherein the first index and the second index identify a same end of the array. Comer has taught wherein the first index and the second index identify a same end of the array (Comer pages 41-43). It would have been obvious to a person of ordinary skill in the art to incorporate the first index and second index of Comer, <u>because</u> the beginning and end of the array must be tracked in order not to lose the array and to know where to insert and remove elements. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the first index and second index of Comer in the device of Arnold.

63. Referring to claim 43, Arnold has taught an apparatus comprising:

a. Plural processors (Arnold page 2, line 1);

b. A store addressable by each of the plural processors (Arnold page 2);

c. First- and second-end index stores accessible to each of the plural processors for identifying opposing ends of a bounded-size contiguous array encoded in circular buffer form in the addressable store (Arnold pages 2-3); and

d. Means for coordinating competing access operations, the coordinating means employing in each instance thereof, at least one double compare and swap (DCAS) operation to disambiguate a retry state and a state of the array based on then-current contents of one, but not both, of first- and second-end index stores and an array element corresponding thereto (Arnold pages 2-3).

64. Arnold has not taught detecting a boundary condition state of the double-ended data structure. Comer has taught detecting a boundary condition state of the double-ended data structure (Comer pages 41-42). It would have been obvious to a person of ordinary skill in the art to incorporate the detection of Comer, <u>because</u> in a double-ended data structure boundary conditions must be monitored so an error does not occur. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the detection of Comer in Arnold to prevent errors.

Conclusion

65. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure as follows. Applicant is reminded that in amending in response to a rejection of claims, the patentable novelty must be clearly shown in view of the state of the art disclosed by

the references cited and the objections made. Applicant must also show how the amendments avoid such references and objections. See 37 CFR § 1.111(c).

a. Brown et al., U.S. Patent Number 3,886,525, has taught compare and swap instructions and its uses.

b. James P. Cohoon and Jack W. Davidson's <u>C++ Program Design: An Introduction to Programming and Object-Oriented Design Second Edition</u> ©1999 pages 465-502 has taught lists, arrays, and queues and basic implementations and methods associated with them

c. Bjarne Stroustrup's <u>The C++ Programming Language Third Edition</u> ©1997 pages 461-497 has taught list, stack, and queue implementations.

66. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aimee J Li whose telephone number is (703) 305-7596. The examiner can normally be reached on M-T 7:30am-5:00pm.

67. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone numbers for the organization where this application or proceeding is assigned are (703) 746-7239 for regular communications and (703) 746-7238 for After Final communications.

68. Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Aimee J. Li
Examiner
Art Unit 2183

April 3, 2003

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100